

# Grid and Zone Selection for AMR and ALE Schemes

*L. Jameson, J. Johnson, B. Bihari, D. Eliason, T. Peyser*

**September 11, 2002**

**U.S. Department of Energy**

Lawrence  
Livermore  
National  
Laboratory

## DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U. S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

This report has been reproduced directly from the best available copy.

Available electronically at <http://www.doc.gov/bridge>

Available for a processing fee to U.S. Department of Energy  
And its contractors in paper from  
U.S. Department of Energy  
Office of Scientific and Technical Information  
P.O. Box 62  
Oak Ridge, TN 37831-0062  
Telephone: (865) 576-8401  
Facsimile: (865) 576-5728  
E-mail: [reports@adonis.osti.gov](mailto:reports@adonis.osti.gov)

Available for the sale to the public from  
U.S. Department of Commerce  
National Technical Information Service  
5285 Port Royal Road  
Springfield, VA 22161  
Telephone: (800) 553-6847  
Facsimile: (703) 605-6900  
E-mail: [orders@ntis.fedworld.gov](mailto:orders@ntis.fedworld.gov)  
Online ordering: <http://www.ntis.gov/ordering.htm>

OR

Lawrence Livermore National Laboratory  
Technical Information Department's Digital Library  
<http://www.llnl.gov/tid/Library.html>

# Grid and Zone Selection for AMR and ALE Schemes

Leland Jameson, John Johnson, Barna Bihari

Don Eliason and Tom Peyser

Lawrence Livermore National Laboratory  
University of California  
P.O. Box 808, MS L-039  
Livermore, CA 94551  
Email: jameson3@llnl.gov

## ABSTRACT

Numerical algorithms are based fundamentally on polynomial interpolation. In regions of the computational domain where a low order polynomial fits the data well one will find small errors in the computed quantities. Therefore, in order to design robust methods for grid selection for AMR schemes or zone selection for ALE schemes, one needs some information on the local polynomial structure of the fields being computed. We provide here algorithms and software for selecting zones based on local estimates of polynomial interpolation error. The algorithms are based on multiresolution and wavelet analysis.

**Key Words:** Wavelet Analysis, Multiresolution Analysis, Grid Generation, AMR, ALE

This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

# 1 Introduction

Traditionally, Adaptive Mesh Refinement (AMR) algorithms use grid refinement and coarsening flags which rely on a combination of information from the first derivative and the second derivative operators. Such flags usually work, but one can always construct counter-examples when they fail. What one needs instead is a direct measure of how well the computational data fits a local polynomial. In other words, one needs a measure of the local truncation error of the interpolation since this is exactly what produces the errors in the numerical operators. Multiresolution and wavelet analysis can provide these estimates of local truncation error in a very cost effective manner.

Alternating Lagrangian-Eulerian Schemes basically follow the mass and the zone selection is, therefore, determined by where the mass goes and by where the remap process places zones that have tangled or that will tangle soon. The strong point of the ALE approach is that one will maintain grid points on material interfaces and a weak point is that following the mass does not produce grids that keep polynomial truncation error low. Therefore, one can have clusters of grid points in regions where mass has congregated producing density profiles that are large in magnitude but more or less uniform. But, from a numerical point of view, one does not need to have zones clustered where mass is clustered, one needs zones where mass transitions from low density profiles to high density profiles and visa versa. Likewise, one needs zones where any of the computation variables "change". Where, "change" is quantified by how well the grid point values of the given variable fit a local low-order polynomial.

The methods that we are suggesting here can be used in conjunction with existing grid and zone selection algorithms. For example, with ALE schemes one might consider weighting the zone selection mechanism so that truncation error is kept within reasonable limits, thereby keeping the numerical errors under control. If there are regions of the computational domain where refinement alone can not resolve the features, then one can use wavelets to do local filtering do damp out oscillations or other features that are considered to be of numerical origin or simply unwanted for stability considerations, see [7].

This document is structured as follows: First we explain a bit of wavelet analysis and multiresolution analysis since this is the theoretical framework on which the grid selection is based. Next, we review the fundamental issues

in the construction of numerical schemes. Finally, we give Fortran software that does the actual grid selection. We plan to construct similar C software if sufficient interest exists.

## 2 Wavelets and Polynomial Structure

In order to understand the grid refinement and zone selection mechanism, we will need a brief introduction to multiresolution and wavelet analysis. Note that the current usage of multiresolution is in the spirit of Harten, see [2], [3], and applied in many areas of application, see [4], [5]. In Harten's approach, one finds polynomial truncation error by examining the difference between interpolated values on a grid of  $N$  points with the exact values provided on a grid of  $2N$  points. The information provided is then used to decide when should call expensive flux evaluations and when a simple and very inexpensive interpolated value is sufficient. From here our focus will be on the orthogonal class of wavelets known as Daubechies wavelets and the applications to grid selection, see [1], [10], [11], [12], [13], and [14]. But, we consider the two approaches, Harten's multiresolution and the traditional wavelet approach, to be more or less equivalent for the purposes of selecting numerical grids and zones.

To define Daubechies-based wavelets, see [9] for the original work, consider the two functions  $\phi(x)$ , the scaling function, and  $\psi(x)$ , the wavelet. The scaling function is the solution of the dilation equation,

$$\phi(x) = \sqrt{2} \sum_{k=0}^{L-1} h_k \phi(2x - k), \quad (1)$$

where  $\phi(x)$  is normalized  $\int_{-\infty}^{\infty} \phi(x) dx = 1$ , and the wavelet  $\psi(x)$  is defined in terms of the scaling function,

$$\psi(x) = \sqrt{2} \sum_{k=0}^{L-1} g_k \phi(2x - k). \quad (2)$$

One builds an orthonormal basis from  $\phi(x)$  and  $\psi(x)$  by dilating and translating to get the following functions:

$$\phi_k^j(x) = 2^{-\frac{j}{2}} \phi(2^{-j}x - k), \quad (3)$$

and

$$\psi_k^j(x) = 2^{-\frac{j}{2}}\psi(2^{-j}x - k), \quad (4)$$

where  $j, k \in Z$ .  $j$  is the dilation parameter and  $k$  is the translation parameter. The coefficients  $H = \{h_k\}_{k=0}^{L-1}$  and  $G = \{g_k\}_{k=0}^{L-1}$  are related by  $g_k = (-1)^k h_{L-k}$  for  $k = 0, \dots, L-1$ . All wavelet properties are specified through the parameters  $H$  and  $G$ . If one's data is defined on a continuous domain such as  $f(x)$  where  $x \in R$  is a real number then one uses  $\phi_k^j(x)$  and  $\psi_k^j(x)$  to perform the wavelet analysis. If, on the other hand, one's data is defined on a discrete domain such as  $f(i)$  where  $i \in Z$  is an integer then the data is analyzed, or filtered, with the coefficients  $H$  and  $G$ . In either case, the scaling function  $\phi(x)$  and its defining coefficients  $H$  detect localized low frequency information, i.e., they are low-pass filters (LPF), and the wavelet  $\psi(x)$  and its defining coefficients  $G$  detect localized high frequency information, i.e., they are high-pass filters (HPF). Specifically,  $H$  and  $G$  are chosen so that dilations and translations of the wavelet,  $\psi_k^j(x)$ , form an orthonormal basis of  $L^2(R)$  and so that  $\psi(x)$  has  $M$  vanishing moments which determines the accuracy. In other words,  $\psi_k^j(x)$  will satisfy

$$\delta_{kl}\delta_{jm} = \int_{-\infty}^{\infty} \psi_k^j(x)\psi_l^m(x)dx, \quad (5)$$

where  $\delta_{kl}$  is the Kronecker delta function, and the accuracy is specified by requiring that  $\psi(x) = \psi_0^0(x)$  satisfy

$$\int_{-\infty}^{\infty} \psi(x)x^m dx = 0, \quad (6)$$

for  $m = 0, \dots, M-1$ . Under the conditions of the previous two equations, for any function  $f(x) \in L^2(R)$  there exists a set  $\{d_{jk}\}$  such that

$$f(x) = \sum_{j \in Z} \sum_{k \in Z} d_{jk} \psi_k^j(x), \quad (7)$$

where

$$d_{jk} = \int_{-\infty}^{\infty} f(x)\psi_k^j(x)dx. \quad (8)$$

For Daubechies wavelets the number of coefficients in  $H$  and  $G$ , or the length of the filters  $H$  and  $G$ , denoted by  $L$ , is related to the number of vanishing moments  $M$  by  $2M = L$ . The coefficients  $H$  needed to define compactly supported wavelets with a higher degree of regularity can be found in

[9]. As is expected, the regularity increases with the support of the wavelet. The usual notation to denote a Daubechies-based wavelet defined by coefficients  $H$  of length  $L$  is  $D_L$ .

It is usual to let the spaces spanned by  $\phi_k^j(x)$  and  $\psi_k^j(x)$  over the parameter  $k$ , with  $j$  fixed, be denoted by  $V_j$  and  $W_j$  respectively,

$$V_j = \text{span}_{k \in \mathbb{Z}} \phi_k^j(x), \quad (9)$$

$$W_j = \text{span}_{k \in \mathbb{Z}} \psi_k^j(x). \quad (10)$$

The spaces  $V_j$  and  $W_j$  are related by,

$$\dots \subset V_1 \subset V_0 \subset V_{-1} \subset \dots, \quad (11)$$

and

$$V_j = V_{j+1} \oplus W_{j+1}. \quad (12)$$

The previously stated condition that the wavelets form an orthonormal basis of  $L^2(R)$  can now be written as,

$$L^2(R) = \bigoplus_{j \in \mathbb{Z}} W_j. \quad (13)$$

## 2.1 Restriction to Finite Dimensions

Of course, infinite sums are meaningless when one begins to implement wavelet analysis on a computer where there is always a largest scale, a smallest scale, as well as boundaries. That is, one must limit the range of the scale parameter  $j$  and the location parameter  $k$ . The location parameter  $k$  can be limited by, say, imposing periodic boundary conditions which would require that  $k$  also be periodic or by building special scaling functions and wavelets at the boundaries, see [8], [16].

Consider now the scale parameter  $j$ . As stated above, the wavelet expansion is complete in the sense that an arbitrary function with finite energy can be represented by 'summing up' the orthogonal subspaces  $W_j$  which contain frequency components related to the parameter  $j$ :  $L^2(R) = \bigoplus_{j \in \mathbb{Z}} W_j$ . Therefore, any  $f(x) \in L^2(R)$  can be written as,

$$f(x) = \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} d_k^j \psi_k^j(x).$$

In this expansion, functions with arbitrarily small-scale structures can be represented. In practice, however, there is a limit to how small the smallest structure can be. This would depend, for example, on how fine the grid is in a numerical computation scenario or perhaps what the sampling frequency is in a signal processing scenario. Therefore, on a computer an expansion would take place in a space such as,

$$V_0 = W_1 \oplus W_2 \oplus \dots \oplus W_J \oplus V_J, \quad (14)$$

and would appear as,

$$P_{V_0}f(x) = \sum_{k \in Z} s_k^J \phi_k^J(x) + \sum_{j=1}^J \sum_{k \in Z} d_k^j \psi_k^j(x), \quad (15)$$

where, again,  $d_k^j = \int_{-\infty}^{\infty} f(x) \psi_k^j(x)$ , and  $s_k^J = \int_{-\infty}^{\infty} f(x) \phi_k^J(x)$ . In this expansion, scale  $j = 0$  is arbitrarily chosen as the finest scale that is needed, and scale  $J$  would be the scale at which a kind of local average,  $\phi_k^J(x)$ , provides sufficient large scale information.

### 3 Estimating Computational Errors

As outlined in the previous section, scaling functions are designed to approximate low order polynomials exactly up to a given order, and wavelets are orthogonal to these same low-order polynomials. Any deviation from low-order polynomial structure in a computational domain can then be detected by wavelet analysis, see Jameson 1998. This measure of deviation from low-order polynomial structure is exactly what is needed to measure computational error. The reason for this is that fundamentally all non-spectral numerical schemes are constructed from low-order algebraic polynomials, and



such schemes are exact if the data falls exactly on a low-order polynomial. Actually, spectral methods also follow this same rule, but in the case of Fourier spectral methods, the polynomials are trigonometric and not algebraic and are global instead of local. To be precise, let us review the fundamentals of how numerical methods are constructed and how wavelet analysis can detect errors in a computational scheme.

Numerical schemes for the approximation of partial differential equations on a computer provide a mechanism for taking one set of  $N$  numbers to another set of  $N$  numbers. To make this transition from one set to the next set, we must take derivatives but the data is a set of points which are not connected. We must, therefore, choose some type of function with which we can connect these points so that we can take a derivative. There are many choices which can be made, but fundamentally we are always working with some type of polynomial be it algebraic, trigonometric, or other.

Simply said, one can generate differencing coefficients by first interpolating a polynomial of any order through a set of data, followed by differentiation of this polynomial and evaluated at a grid point. As long as the number of grid points exceeds the order the polynomial by one, then the interpolation is unique and the differencing coefficients are likewise unique.

Interpolation with algebraic polynomials is probably the most common form of interpolation, and it is from this type of interpolation that common uniform grid finite difference methods can be found. The well-known Lagrange interpolation formula for algebraic interpolation is,

$$A_j(x) = \prod_{k=0, k \neq j}^n (x - x_k) / \prod_{k=0, k \neq j}^n (x_j - x_k). \quad (16)$$

$A_j(x_k) = \delta_{jk}$  For given values  $w_0, w_1, \dots, w_n$ , the polynomial

$$p_n(x) = \sum_{k=0}^n w_k A_k(x). \quad (17)$$

in  $P_n$  and takes on these values at the points  $x_i$ :

$$p_n(x_k) = w_k, \quad (18)$$

for  $k = 0, 1, \dots, n$ , and it is the truncation error of this interpolation which determines the order accuracy that one obtains when differentiating.

Recall that the remainder for algebraic polynomial interpolation is,

$$f(x) - p_n(x) = \frac{(x - x_0)(x - x_1)\dots(x - x_n)}{(n + 1)!} f^{(n+1)}(\xi), \quad (19)$$

where  $\xi$  lies between the smallest and the largest  $x_i$ . Note that when one builds differencing operators after interpolation that the term,

$$f^{(n+1)}(\xi), \quad (20)$$

is unchanged whereas the term involving the products of the point of interest  $x$  and the grid points  $x_i$  goes from a term of the form,

$$\prod_{i=0}^n (x - x_i) \quad (21)$$

to a term of the form,

$$\prod_{i=0}^{n-1} (x - x_i). \quad (22)$$

In other words, if the computational data is exactly a polynomial of order  $n$  or less then the remainder term is zero and there is no error in the computation.

Consider the simple case of interpolating an algebraic quadratic polynomial  $p_2(x)$  to a function  $f(x)$  at the grid points  $x_0 < x_1 < x_2$ :  $p_2(x_i) = f(x_i)$ ,  $i = 0, 1, 2$ . The remainder term for some  $\xi$ ,  $x_0 \leq \xi \leq x_2$ , is

$$f(x) - p_2(x) = (x - x_0)(x - x_1)(x - x_2) \frac{1}{3!} f^{(3)}(\xi). \quad (23)$$

Now, differentiate and evaluate at  $x = x_1$  to get,

$$f'(x_1) - p_2'(x_1) = (x_1 - x_0)(x_1 - x_2) \frac{1}{3!} f^{(3)}(\xi) = Ch^2 f^{(3)}(\xi), \quad (24)$$

where  $h = x_1 - x_0 = x_2 - x_1$ . If the grid is evenly-spaced then the differences  $(x_i - x_j)$  are some integer multiple of the smallest difference which one can denote by  $h$ . If one doubles the number of grid points then each of the distances  $(x_i - x_j)$  becomes half as large and the accuracy for this quadratic example will be 2. To be specific, if the computational data is quadratic polynomial or less, then the remainder is zero and the computation has no error.

In practice, however, computational data will rarely be an exact low-order polynomial, therefore there is always error. The size of this computational error will depend on the deviation from low-order polynomials and can be readily measured with wavelet analysis. If on the other hand, the data is exactly a low-order polynomial and the numerical method is "exact", then wavelets of the corresponding order of accuracy will be orthogonal to the data and all wavelet coefficients will be zero indicating, correctly, that the calculation has no error.

Recall from above that the wavelet coefficients are found as the inner product of the function at hand and the corresponding wavelet,

$$d_k^j = \int_{-\infty}^{\infty} f(x) \psi_k^j(x) dx , \quad (25)$$

where the decay of the expansion coefficients depends on the local regularity of  $f(x)$  and one can derive the following expression, see Strang 1996:

$$|d_k^j| \leq C 2^{-j \frac{L+1}{2}} \max_{\xi \in [k2^{-j}, (k+M-1)2^{-j}]} |f^{(M)}(\xi)| . \quad (26)$$

From Eq.(26) we find that if  $f(x)$  behaves like a polynomial of order less than  $M$  inside the small interval, then  $d_k^j$  vanishes exactly.

Thus, by considering the magnitude of  $d_k^j$  one obtains a local measure of the deviation of the computational data from low-order polynomials. Indeed, the information given by Eq.(26) is of very local character and provides an excellent estimate of local calculation error. In Figures (1), (2), (3), and (4).

## 4 Software Implementation

In this section we present the actual software that performs the grid and zone selection mechanism. The main subroutine *newgr.f* performs two major functions. The lines of code from line 9 to line 13 perform the wavelet analysis producing the wavelet coefficients of the input function  $f_i$ , and from line 15 to line 43 the numerical grid is generated based on these wavelet coefficients.

### 4.1 Generating the Wavelet Coefficients: lines 9-13

To illustrate, assume that the user has determined that 8 is the maximum desired ratio between the maximum  $\Delta x$  and the minimum  $\Delta x$ . As noted

above, this corresponds to 3 wavelet decompositions. As above, let  $V_0$  denote the subspace spanned by scaling functions on the finest scale. One wavelet decomposition produces the division of  $V_0$  into  $V_1$  and  $W_1$ :  $V_0 = W_1 \oplus V_1$ . Similarly, three wavelet decompositions produces,  $V_0 = W_1 \oplus W_2 \oplus W_3 \oplus V_3$ . Each of these wavelet decompositions is performed by the subroutine *filter.f*. The input to *filter.f* is the variable *Extdata* which contains the scaling function coefficients for subspace  $V_i$  and the output variables of *filter.f* are the variables *data* and *HPF* which contain the coefficients for the subspaces  $V_{i+1}$  and  $W_{i+1}$ , respectively. The variable is named *Extdata* because the coefficients of  $V_i$  have been ‘extended’ to reflect the boundary conditions. In this version of the program, the *data* is extended by adding constant scaling function coefficient values to the ends of the vector *data* by the routine *constext.f*. Note, if one desires periodic boundary conditions then one ‘wraps’ the scaling function coefficients around such that one extends the vector *data* by returning to the beginning of the same vector. Likewise, if one wants a smoother extension of *data* then one can write a routine which extends linearly or by some other higher order polynomial.

## 4.2 Generating the Grid: lines 15-43

The lines of code from 16 to 20 add the grid points which are referred to as the base grid. That is, these points are evenly-spaced and depend only on the number of wavelet decompositions one has chosen. For example, if  $N = 128$  and  $Nd = 3$  then the base grid will consist of  $16 = 128/2^3$  evenly-spaced points. If the input function  $f$  is smooth with respect to the threshold,  $th$ , then the output grid  $x_0$  could very well be composed only of the base grid.

The lines of code from 21 to 42 add the wavelet refinement to the base grid. Using the standard wavelet notation for the example with 3 decompositions,  $V_0 = W_1 \oplus W_2 \oplus W_3 \oplus V_3$ , the base grid corresponds to the scaling function subspace  $V_3$  in which all the scaling functions are used. If the spacing between grid points in  $V_0$  is  $\Delta x$  then the spacing between grid points in  $V_3$  will be  $8\Delta x$ . The addition of the wavelets in  $W_3$  which have coefficients larger in magnitude than the threshold  $th$  will refine the grid to a spacing of  $4\Delta x$  in these large coefficient regions. Likewise, adding  $W_2$  refines to  $2\Delta x$  and adding  $W_1$  refines to the finest scale of  $\Delta x$ . Testing the magnitude of the wavelet coefficients occurs on line 32 of the code. The variable *iflagpoint* is used with *iw* in order to include a grid point  $x_i(ipnt)$  if, say,  $2/3$  of the wavelet

coefficients in the region around  $xi(ipnt)$  are large. This mechanism adds a kind of ‘softness’ to the grid selection mechanism and appears to work very well when  $iw = 1$ . Line 38 is where the new grid  $xo$  is constructed. Lines 44 and 45 simply add the right-hand boundary grid point and function value.

## 4.3 The input and output variables

### Input Variables

- $xi$  = The evenly-spaced grid point values.
- $fi$  = The evenly-spaced samples of the function which is to be analyzed.  $fi(1)$  = value at left-hand boundary.  $fi(N+1)$  = value at right-hand boundary. If boundary conditions are periodic,  $fi(1) = fi(N+1)$ .
- $L$  = Defines which wavelet is used. For Daubechies 4,  $L=4$ .
- $N$  = The number of points in  $fi$  minus 1.  $N$  is a power of 2.
- $th$  = Threshold to determine which grid points are used. If  $th \leq 0$  then all grid points are used. If  $th$  = large number, perhaps 10, then only the grid points on the ‘coarsest’ grid are used.
- $Nd$  = Number of wavelet decompositions, e.g., if  $Nd = 3$ , then the ratio of the maximum  $\Delta x$  to the minimum  $\Delta x$  is  $8 = 2^3$ .
- $iw$  = Width of wavelet refinement stencil. If  $iw = 1$ , then the magnitude of wavelet coefficients are checked at three locations from  $-iw$  to  $iw$  or at the locations  $-1, 0, 1$  in order to determine if the grid point at location 0 should be used. So that if one has a hyperbolic system, or traveling waves, then if  $iw > 2$  one can add grid points by looking ‘backwards’ and ‘forwards’ for a perturbation which might move into the region currently being examined. This is a kind of preparation for the future evolution of the system at hand.

### Output Variables

- $xo$  = The new wavelet-chosen grid. Note that the grid points on the boundaries are always used. That is,  $xo(1) = xi(1)$  and  $xo(No) = xi(N+1)$ .

- fo = The function values on the new wavelet-chosen grid.
- No = The number of grid points in the new grid. Note that whereas 'No' counts every grid point, the input variable 'N' does not include the last point on the right hand boundary. This is done to facilitate the use with periodic as well as non-periodic boundary conditions.

## 5 Grid Generation Subroutines

The following four subroutines provide a stand-alone 1 dimensional grid generation and zone selection package.

### 5.1 newgr.f

This is the main subroutine which will be called by the user created driver program.

```

1 subroutine newgr(xi,fi,L,N,th,Nd,iw,xo,fo,No)
2 parameter(Nmax = 260, Lmax = 8, Ndmax = 8)
3 real xi(Nmax),fi(Nmax),xo(Nmax),fo(Nmax),h(Lmax),g(Lmax)
4 real HPF(Nmax/2+Lmax,Ndmax),data(Nmax),th,Extdata(Nmax+Lmax)
5 call getcoef(L,h,g)
6 do i = 1,Nmax
7 data(i) = fi(i)
8 enddo
9 do idecomp = 1, Nd
10 Ndim = N/(2**(idecomp-1))
11 call constext(data,Ndim,L,Extdata)
12 call filter(Extdata,h,g,Ndim,L,data,HPF(1,idecomp))
13 enddo
14 igrd = 0
15 do 10, ipnt = 1,N
16 if ( abs( mod(ipnt-1,2**(Nd))) .LT. .00001 ) then
17 igrd = igrd + 1
18 xo(igrd) = xi(ipnt)
19 fo(igrd) = fi(ipnt)
20 endif
21 do 20, idecomp = 1,Nd

```

```

22 n1 = abs(ipnt - 2**(idecomp-1) - 1)
23 n2 = 2**(idecomp)
24 if ( abs(mod(n1,n2)) .LT. .00001 ) then
25 index1 = 1+nint(real(n1)/real(n2))
26 iflagpoint = 0
27 do iwid = -iw, iw
28 iindex = index1 + iwid
29 if(iindex.LE.1.OR.iindex.GE.N/(2**(idecomp)))then
30 iindex = index1
31 endif
32 if (abs(HPF(iindex,idecomp)).GT.th)then
33 iflagpoint = iflagpoint + 1
34 endif
35 enddo
36 if (iflagpoint .GE. iw+1) then
37 igrd = igrd + 1
38 xo(igrd) = xi(ipnt)
39 fo(igrd) = fi(ipnt)
40 endif
41 endif
42 20 continue
43 10 continue
44 xo(igrd+1) = xi(N+1)
45 fo(igrd+1) = fi(N+1)
46 No = igrd+1
47 return
48 end

```

## 5.2 getcoef.f

This subroutine is called by *newgr.f* and its only function is to get the wavelet coefficients. Included here are the numbers only for the  $D_4$  wavelet. Other wavelet coefficients can be added by the user. The numerical values for the coefficients  $h(:)$  in the following subroutine came from [9].

```

1 subroutine getcoef(L,h,g)
2 parameter (Lmax = 8)
3 real h(Lmax), g(Lmax)
4 h(1) = .482962913145
5 h(2) = .836516303738

```

```

6 h(3) = .224143868042
7 h(4) = -.129409522551
8 do i = 1,L
9 h(i) = h(i)/(sqrt(2.0))
10 enddo
11 do i = 1,L
12 g(i) = (-1)**(i-1) * h(L - i + 1)
13 enddo
14 return
15 end

```

### 5.3 filter.f

This subroutine does the actual wavelet filtering by dividing *Extdata* into its *high* and *low* components.

```

1 subroutine filter(Extdata,h,g,N,L,low,high)
2 parameter (Nmax = 260,Lmax = 8)
3 real low(Nmax/2+Lmax), high(Nmax/2+Lmax)
4 real Extdata(Nmax+Lmax), h(Lmax), g(Lmax)
5 do i = 1, Nmax/2+Lmax
6 low(i) = 0.0
7 high(i) = 0.0
8 enddo
9 do i = 1, N/2 + (L-2)/2
10 do j = 1,L
11 ij = 2*(i-1) + j - (L-2)
12 low(i) = low(i) + h(j) * Extdata(ij+2)
13 high(i) = high(i) + g(j) * Extdata(ij+2)
14 enddo
15 enddo
16 return
17 end

```

### 5.4 context.f

This routine takes care of the boundary conditions by extending the scaling function coefficients in an appropriate way. The routine provided here extends with constant values. The user can define other routines for whatever boundary conditions are needed.



```

1 subroutine context(data,N,L,Extdata)
2 parameter (Nmax = 260,Lmax = 8)
3 real data(Nmax), Extdata(Nmax+Lmax)
4 do i = 1, N
5   Extdata(L/2+i-1) = data(i)
6 enddo
7 do i = 1, L-3
8   Extdata(L/2-i) = data(1)
9 enddo
10 do i = 1, L-1
11   Extdata(N+L/2+i-1)=data(N)
12 enddo
13 return
14 end

```

## 6 Doubling the Grid Density

Note that the software included here takes a fine grid,  $V_0$ , and chooses from  $V_0$  a subset of points from  $W_1 \oplus W_2 \oplus W_3 \oplus V_3$  to obtain a numerical grid. It is possible that during a numerical simulation that even the finest grid  $\Delta x$  in  $V_0$  is not fine enough and that a grid spacing  $\Delta x/2$  is needed. This is possible by adding to  $V_0$  the refinement  $W_0$  to get  $V_{-1} = V_0 \oplus W_0$ . In the code this can be accomplished by first testing the magnitude of the wavelet coefficients in the subspace  $W_1$  by adding a test statement similar to line 32 in which the magnitude of the numbers in  $\text{HPF}(:,1)$ , corresponding to  $W_1$ , are tested against a second threshold number. For example, if  $th=.001$  then one might decide to double the grid density if magnitude of  $\text{HPF}(:,1) > .01$  (The readers should experiment with these numbers). If this test is true then exit *newgr.f* and double of the grid density of  $xi$  by interpolation thereby making  $N$  become  $2N$  followed by another call to *newgr.f*. Note that it is necessary to double the grid before the data becomes too 'rough'. Once numerical oscillation has begun, it is too late. You must make your threshold numbers sensitive enough to 'see' to high frequency regions coming and refine ahead of time. This type of intuition is easy to develop with a little practice.

## 7 conclusion

In this manuscript we have introduced the use of multiresolution and wavelet analysis for selecting suitable grids for AMR schemes and for zone selection for ALE schemes. The fundamental principle behind using the MR and wavelet approach is that they provide excellent estimates of the local truncation error of the numerical approximations. It is this local truncation error that is responsible for the numerical errors and, thus, one can greatly increase the reliability of numerical simulations by keeping the numerical errors more or less uniform across the computational domain and MR-wavelet analysis provide a very effective tool for this application.

## References

- [1] K. Arai and L. Jameson, "Wavelet Theory and Applications to Global Climate Change," Morikita Press, 2001.
- [2] A. Harten, "Multiresolution algorithms for the numerical solution of hyperbolic conservation laws," *Comm. Pure Appl. Math.*, XLVIII, pp.1305-1342, 1995.
- [3] B.L. Bihari and A. Harten, "Multiresolution schemes for the numerical solution of 2-D conservation laws, I", *SIAM J. Sci. Comput.*, 18, No.2, pp.315-354, 1997.
- [4] B.L. Bihari and D. Schwendeman, "Multiresolution schemes for the reactive Euler equations," *J. Comp. Phys.*, 154, pp. 197-230, 1999.
- [5] B.L. Bihari, "Solution adaptation on unstructured grids via the multiresolution method, " *Numerical Grid Generation in Computational Field Simulations*, Eds. B.K. Soni, J. Hauser, J.F. Thompson, and P. Eiseman, 2000.
- [6] B.L. Bihari, D.K. Ota, Z.Liu, and S.V. Ramakrishnan, "The multiresolution method on general unstructured meshes," *AIAA Paper 2001-2553*, also to appear in *AIAA Journal*.
- [7] W. Cabot, D. Eliason, and L. Jameson, "A Wavelet Based Dissipation Method for ALE Schemes", *UCRL-ID-140074*, 2000.

- [8] A. Cohen, I. Daubechies, and P. Vial, "Wavelets on the interval and fast wavelet transforms", preprint, 1993
- [9] I. Daubechies, "Orthonormal Basis of Compactly Supported Wavelets", *Comm. Pure Appl. Math.*, **41** (1988), pp. 909-996.
- [10] G. Erlebacher, M. Y. Hussaini, L. Jameson, "Wavelets: Theory and Applications", Oxford University Press, 1996.
- [11] L. Jameson, "A Wavelet-Optimized, Very High Order Adaptive Grid and Order Numerical Method", ICASE Report No. 96-30, and *SIAM J. Sci. Comput.*, Vol. 19, No. 6, pp. 1980-2013, November 1998. L.
- [12] L. Jameson and T. Miyama, "Wavelet Analysis and Ocean Modeling: A Dynamically Adaptive Numerical Method "WOFD-AHO", *Monthly Weather Review*, Volume 128, p. 1536-1548, May 2000.
- [13] L. Jameson, T. Adachi, O. Ukai, T. Yuasa, "Wavelet Based Numerical Methods", *International Journal of Computational Fluid Dynamics*, June, 1998.
- [14] L. Jameson, "Wavelet-based Grid Generation", ICASE Report No. 96-59.
- [15] L. Jameson, "Wavelet and Numerical Methods", PhD Thesis, Division of Applied Mathematics, Brown University, 1993.
- [16] Y. Meyer, "Ondelettes sur l'intervalle", *Revista Matematica Iberoamericana* 7, 1992, pp 115-133.
- [17] G. Strang and T. Nguyen, "Wavelets and Filter Banks", Wellesley-Cambridge Press, 1996.

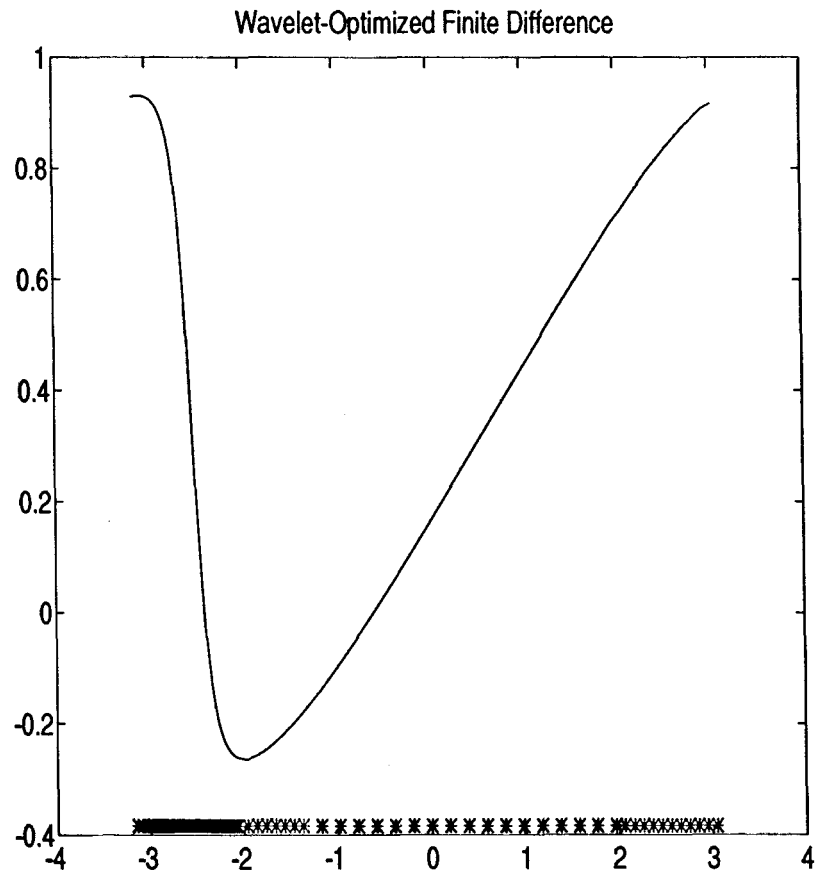


Figure 1: Solution to Burgers equation at a given time with numerical grid as selected by wavelets and multiresolution analysis. Boundary conditions are periodic.

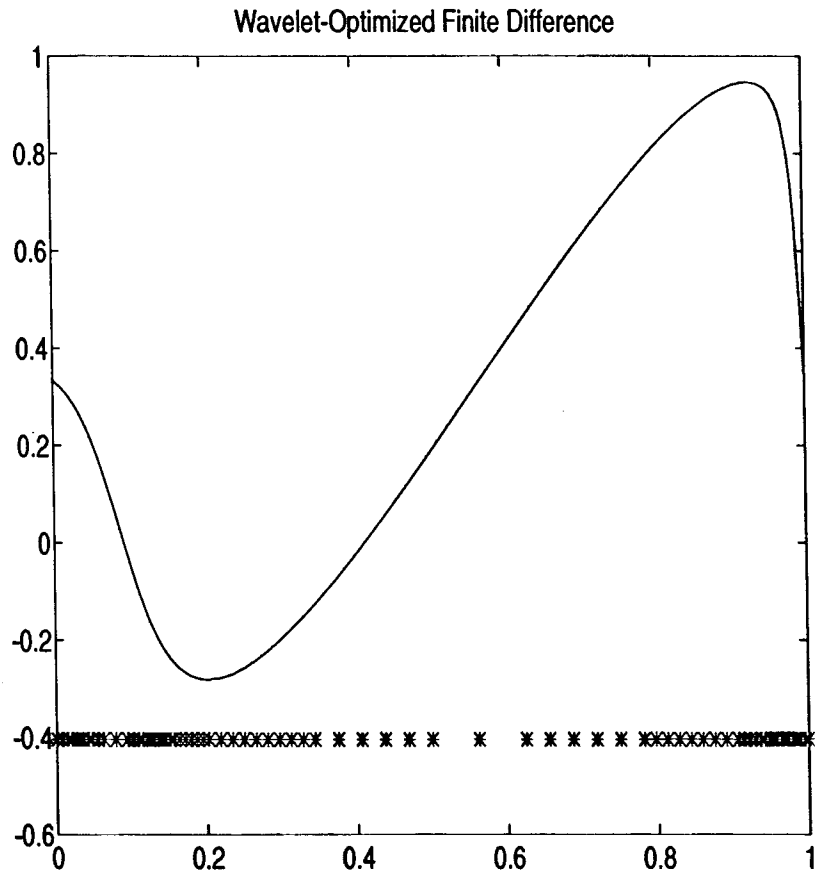


Figure 2: Solution to Burgers equation at a given time with numerical grid as selected by wavelets and multiresolution analysis. Boundary conditions are not periodic.

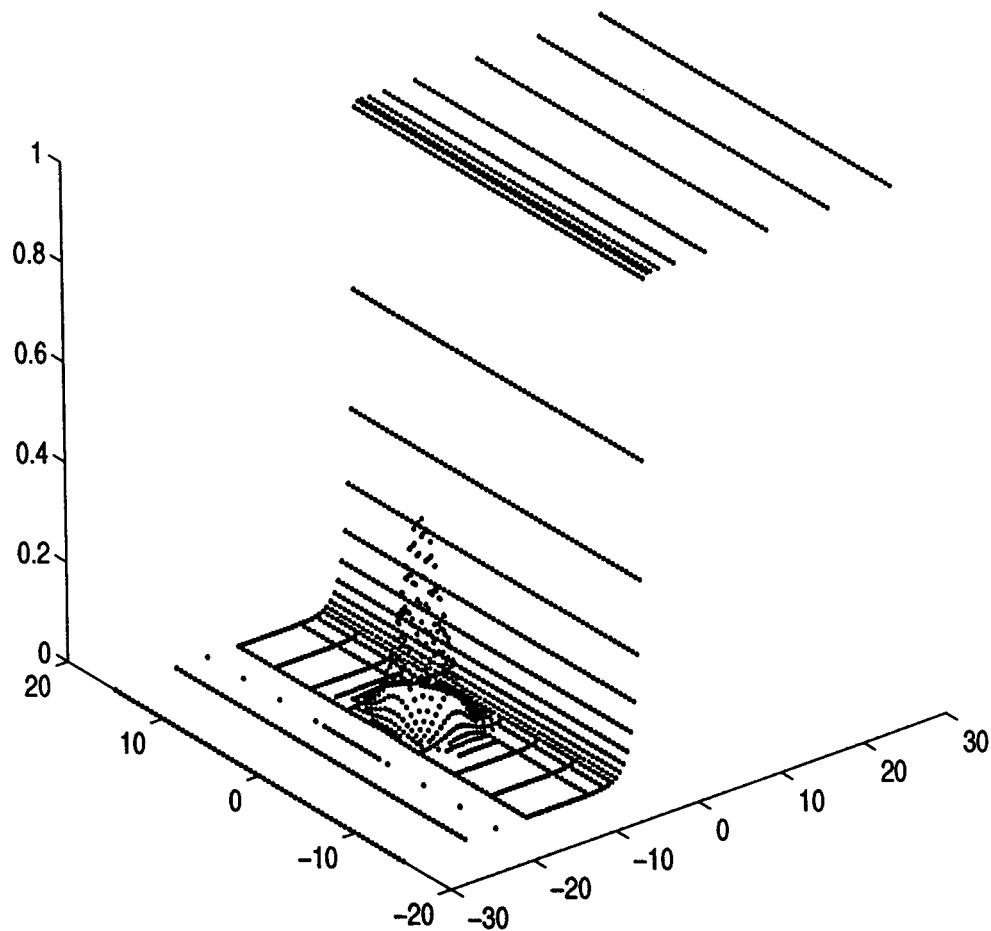


Figure 3: An example of a flame front with steep gradient region requiring adaptive meshing for uniform resolution. Wavelet and Multiresolution analysis give the truncation error necessary for choosing good grids and zones.

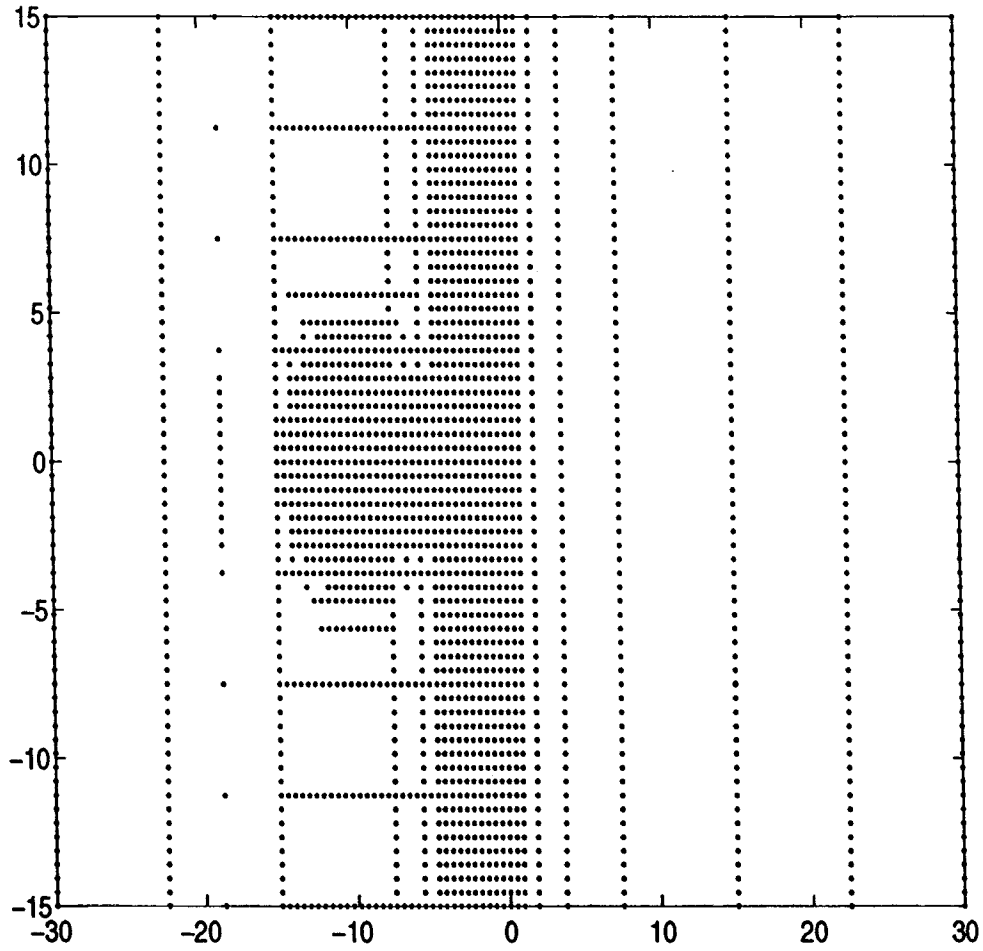


Figure 4: The grid for the flame front as determined by wavelet and multiresolution analysis.